CLAIMS

What is claimed is

1. A register usage indicator system for efficiently signaling register usage in a computer program comprising a plurality of blocks of code, said register usage indicator system comprising:

a code usage register contained within a NOP instruction in one of the plurality of blocks of code in the computer program, said code usage register comprising a plurality of storage bits; and

a code register usage annotator for determining if each one of the plurality of registers is live in one of the plurality of blocks of code containing said NOP instruction.

2. The system of claim 1, wherein said code register usage annotator sets one of said plurality of storage bits in said code usage register for each one of the plurality of registers that is live in one of the plurality of blocks of code containing said NOP instruction.

3. The system of claim 1, further comprising:

a register usage comparator for determining which of said registers are live in one of the plurality of blocks of code in the computer program by inspecting the bits set in said code usage register contained in said NOP instruction.

4. The system of claim 3, wherein said code register usage annotator determines whether or not each register is live in each one of the plurality of blocks of code containing said NOP instruction; and

wherein said code register usage annotator sets each one of the plurality of storage bits in one of a plurality of storage code usage registers for each register live in each one of the plurality of blocks of code containing said NOP instruction.

5. The system of claim 4, wherein said register usage comparator determines which of said registers are not live in one of said plurality of blocks of code, by performing a logical OR of all of said plurality of storage code usage registers.

6. A method to efficiently signal register usage in a computer program comprising a plurality of blocks of code, the method comprising the steps of:

determining which of a plurality of registers are live in one of the plurality of blocks of code in the computer program;

finding at least one NOP instruction in one of the plurality of blocks of code;

creating a code usage register having a plurality of storage bits in said at least one NOP instruction in one of the plurality of blocks of code; and

setting one of said plurality of storage bits for each one of the plurality of registers live in one of the plurality of blocks of code containing said NOP instruction.

B

7. The method of claim 6, wherein said determining step further comprises the step of:

determining which of said plurality of registers are live in one of the plurality of blocks of code by inspecting the bits set in said code usage register.

8. The method of claim 7, further comprising the step of:

determining which of the plurality of registers is live in each one of the plurality of blocks of code in the computer program.

9. The method of claim 8, further comprising the step of:

setting each one of said plurality of storage bits in one of a plurality of storage code usage registers for each register live in one of the plurality of blocks of code containing said NOP instruction.

10. The method of claim 9, further comprising the step of:

determining which of said registers are not live in all of the plurality of blocks of code, in the computer program, by performing a logical OR of all of said plurality of storage code usage.

11. A register usage indicator system for efficiently signaling register usage in a computer program comprising a plurality of blocks of code, said register usage indicator system comprising:

means for determining which of the plurality of registers are live in one of the plurality of blocks of code in the computer program;

means for finding at least one NOP instruction in said one of the plurality of blocks of code;

means for creating a code usage register in said at least one NOP instruction in said one of the plurality of blocks of code; and

means for setting one of a plurality of storage bits in said code usage register for each one of the plurality of registers live in said one of the plurality of blocks of code containing said NOP instruction.

12. The system of claim 11, further comprising:

means for determining which of the plurality of registers are live in each one of the plurality of blocks of code in the computer program.

13. The system of claim 12, wherein said determining means further comprises:

means for inspecting the bits set in said code usage register to determine which of said registers are live in one of the plurality of blocks of code containing said NOP instruction.

14. The system of claim 13, further comprising:

means for determining which of the plurality of registers are live in each one of the plurality of blocks of code in the computer program.

15. The system of claim 14, further comprising:

means for setting each one of said plurality of storage bits in one of a plurality of storage code usage registers for each register live in each one of the plurality of blocks of code in the computer program.

16. The system of claim 15, further comprising:

means for determining which of said registers are not live in any of the plurality of blocks of code in the computer program, by performing a logical OR of all of said plurality of storage code usage registers.

add a1

ADD B1